

(43) Date of A Publication 19.05.1999

(21) Application No 9724076.6

(22) Date of Filing 15.11.1997

(71) Applicant(s)  
**IBM United Kingdom Limited**  
(Incorporated in the United Kingdom)  
**PO Box 41, North Harbour, PORTSMOUTH,**  
**Hampshire, United Kingdom**

(72) Inventor(s)  
**Alexander John Wilson**

(74) Agent and/or Address for Service  
**J D Williams**  
**IBM United Kingdom Limited, Intellectual Property**  
**Department, Mail Point 110, Hursley Park,**  
**WINCHESTER, Hampshire, SO21 2JN,**  
**United Kingdom**

(51) INT CL<sup>6</sup>  
**G06F 3/12 // G06F 11/00**

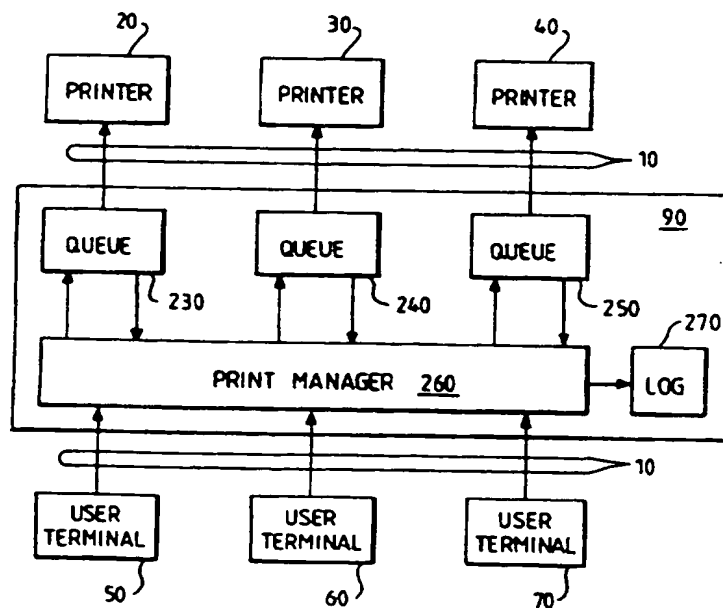
(52) UK CL (Edition Q )  
**G4H HPC HQE H1A H13D H14D**  
**G4A AEX AKB1 A12C A12N**  
**U1S S2247**

(56) Documents Cited  
**None**

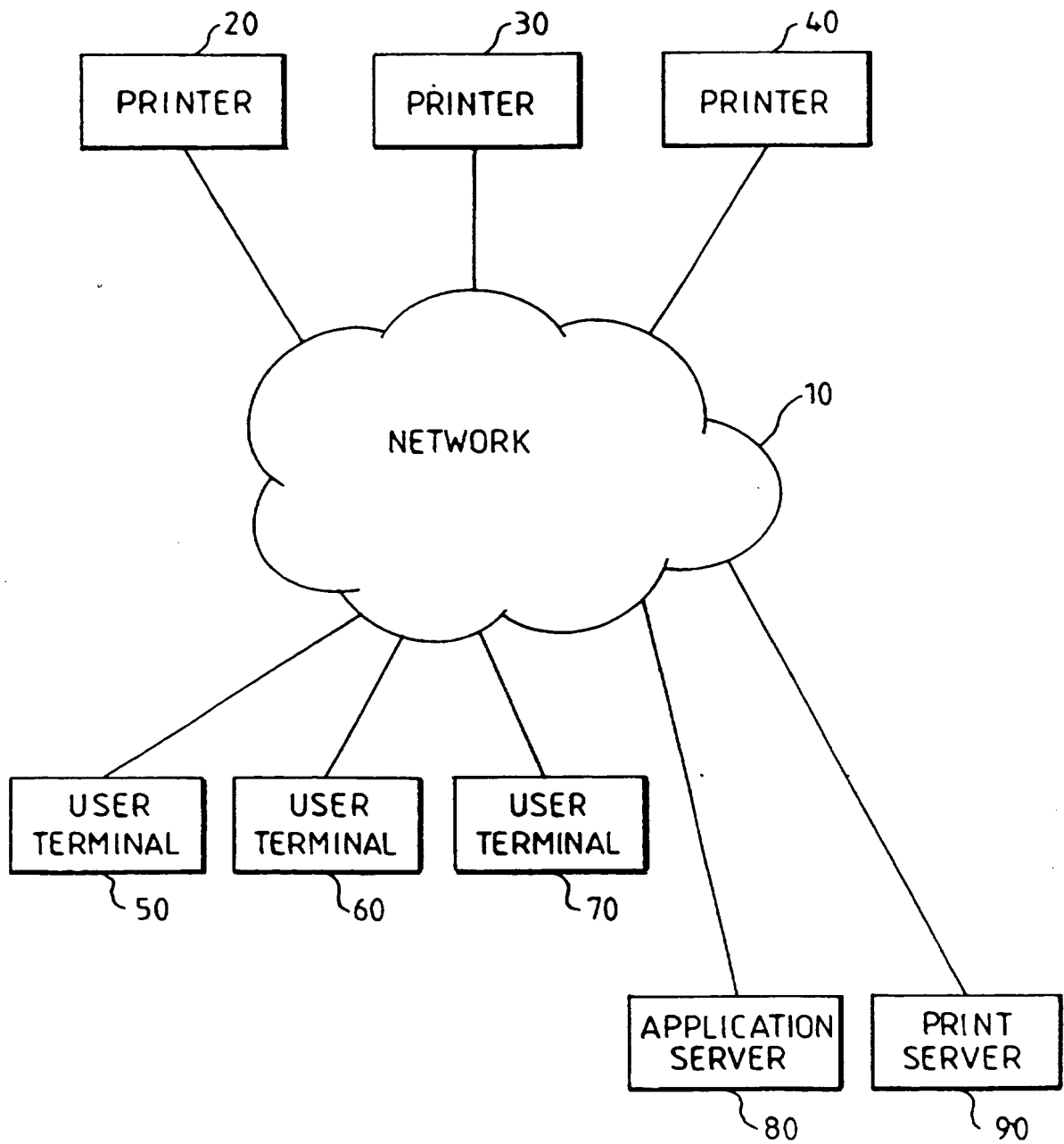
(58) Field of Search  
UK CL (Edition P ) **G4A AKB1 , G4H HPC**  
INT CL<sup>6</sup> **G06F**

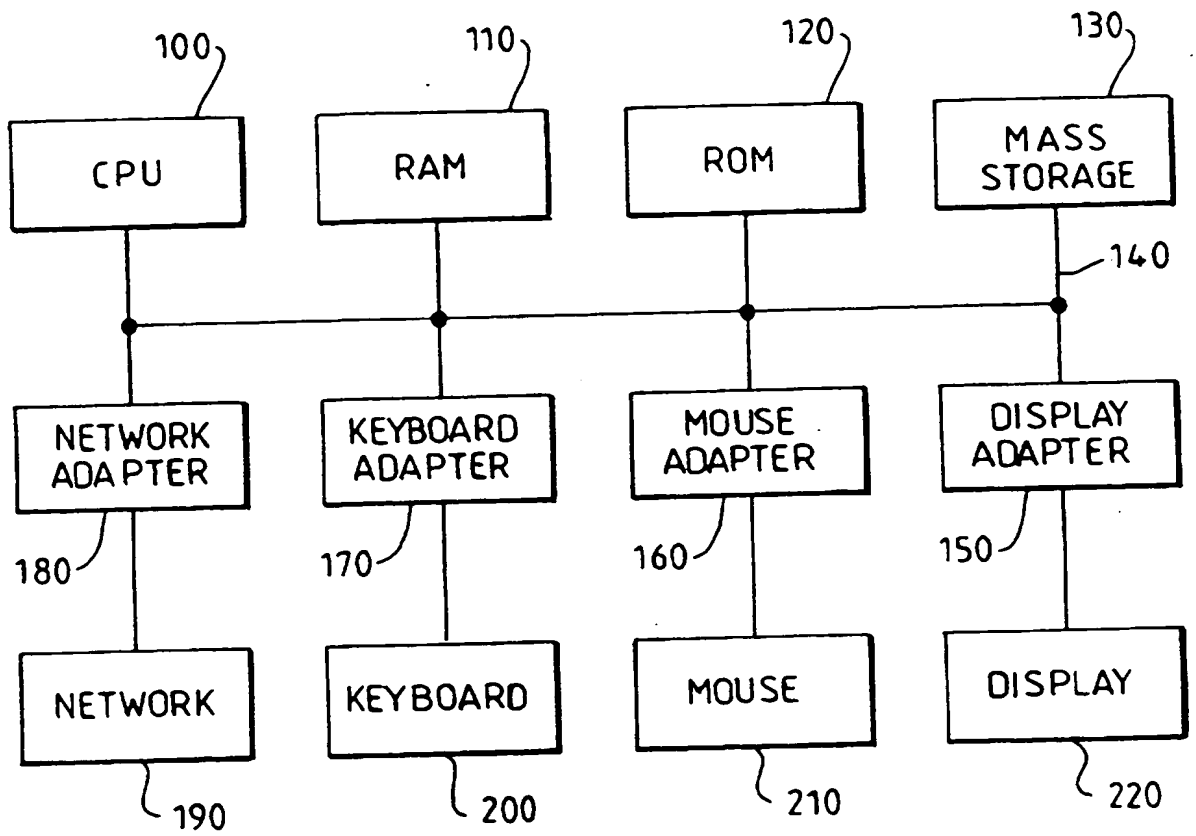
(54) Abstract Title  
**Controlling printers**

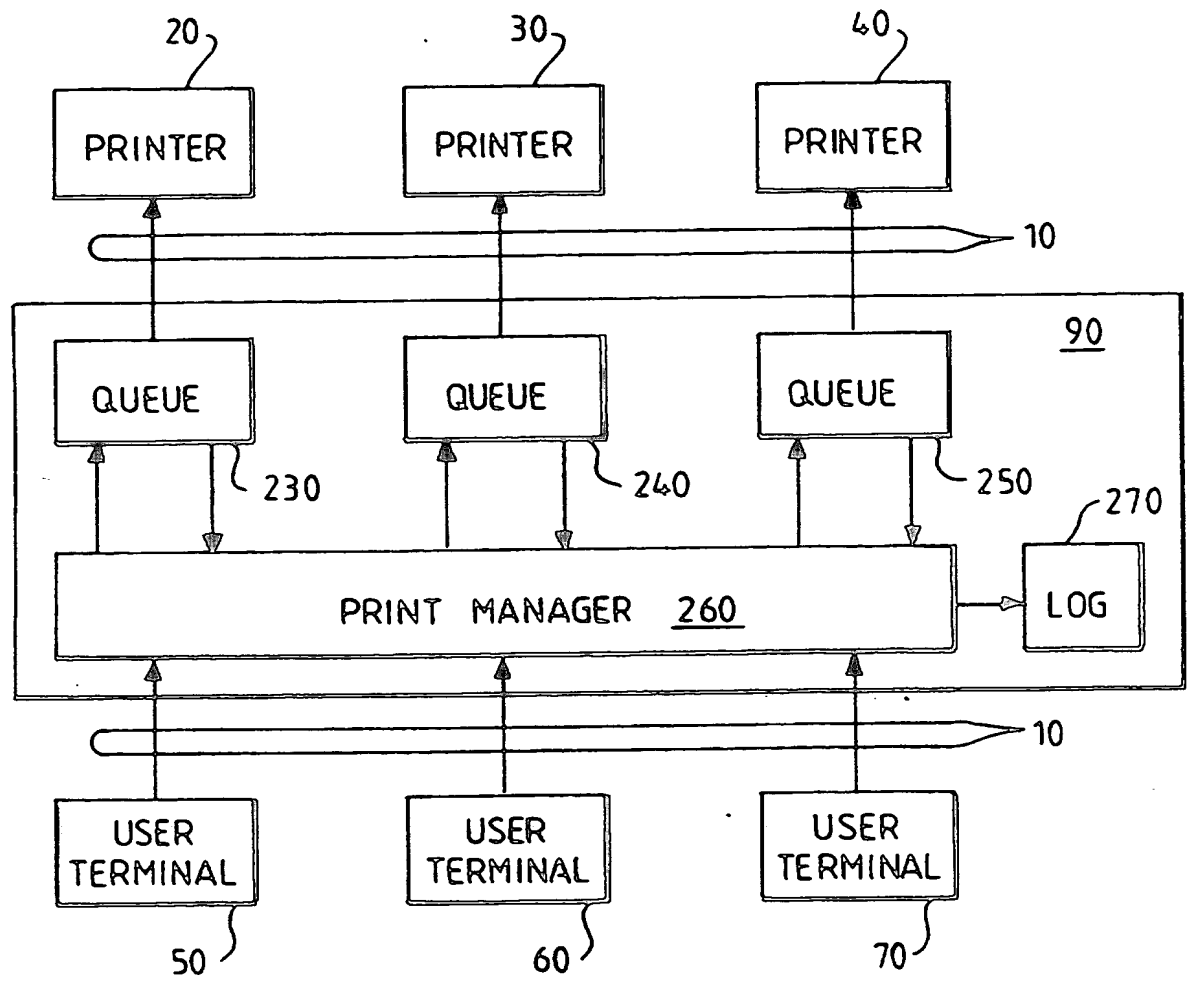
(57) A printer controller for a server computer system 90 of a data processing system comprising a user terminal, a printer, and a network 10 interconnecting the server, the user terminal and the printer, receives print jobs from the user terminal via the network, stores the print jobs in a print queue, sequentially sends the print jobs from the print queue to the printer via the network, and automatically deletes from the print queue print jobs stored in the print queue for a period in excess of a predetermined threshold. The jobs are accompanied by print drivers, and the server detects when the driver does not match the printer and sends an alert or replaces the driver.

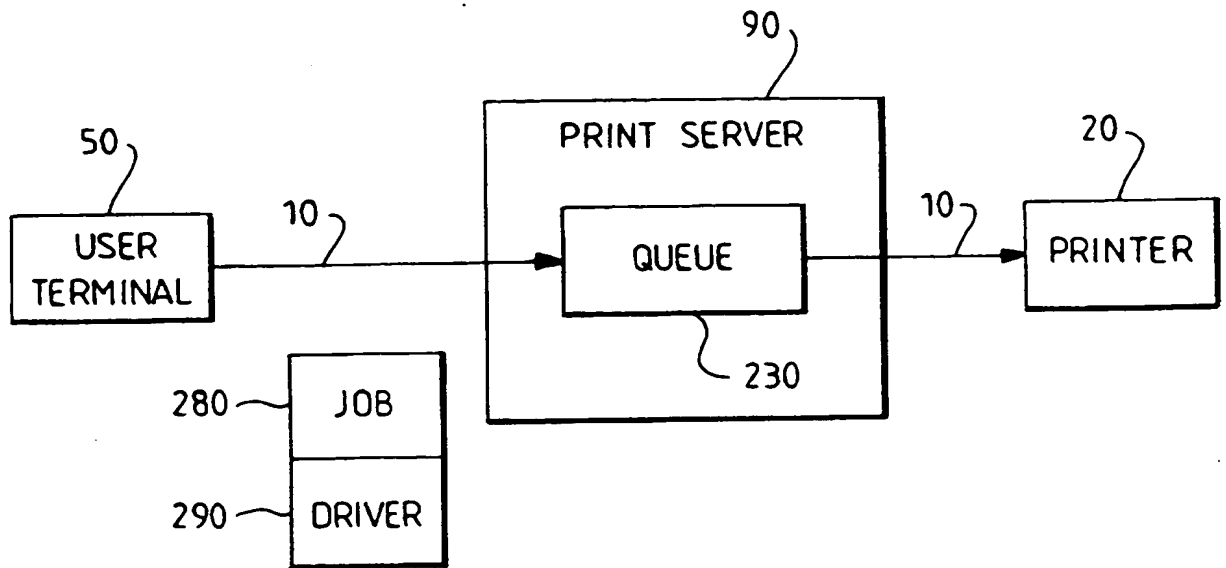


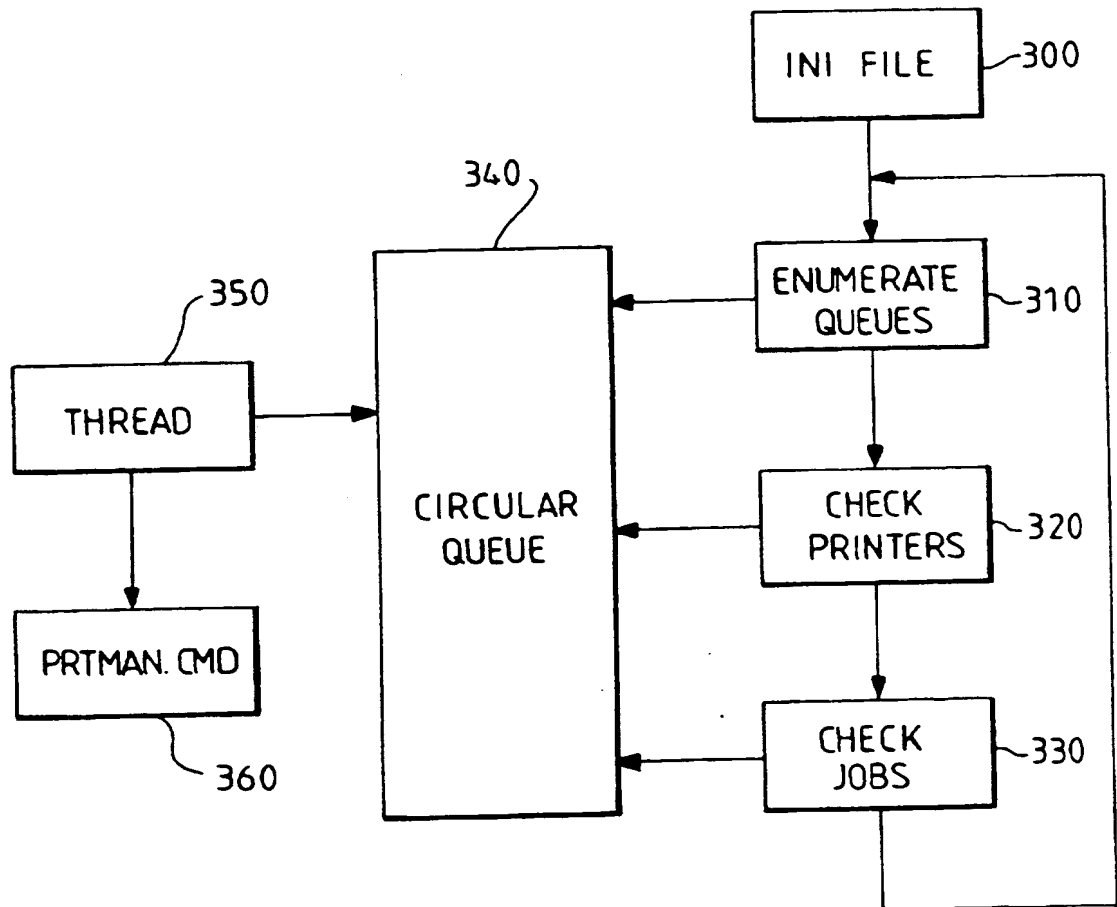
**FIG. 3**

**FIG. 1**

**FIG. 2**

FIG. 3

FIG. 4

FIG. 5

METHOD AND APPARATUS FOR CONTROLLING PRINTERS

The present invention relates to a method and apparatus for controlling printers in a distributed data processing environment.

5 In a typical distributed data processing environment, there are multiple interconnected local area networks (LANs) each having a plurality of connected user terminals. The interconnected LANs may be spread throughout a building or site to form a campus network portion of the distributed data processing environment. Such a data processing environment may comprise multiple campus networks interconnected to form a wide area network (WAN). In general, each LAN in the data processing environment comprises a server computer system for providing application services to user terminals on request. Examples of such applications include word processor and computer aided drawing software packages. A plurality of printers are typically connected to each LAN for printing files produced by connected users via the word-processing and drawing applications. The printers are controlled by print manager software executing either on the server computer system providing the application services or, more commonly, on a separate print server computer system also connected to the LAN. The print manager has a separate queue for each printer. Any jobs sent by users to a printer in the LAN are stored in the corresponding queue prior to being sent sequentially to the printer.

25 In a typical distributed data processing environment, each LAN is furnished with a range of different printers each optimised for different type of printing. Generally, each printer has it's own configuration or driver software which is down-loaded to the printer via the print manager along with each print job from the user community. When a user sends a job to a queue using the wrong driver, the queue for the specified printer becomes jammed because the printer is unable to accept the job. This problem exasperated by printer driver names changing between different operating systems on the user terminals. Users may be unaware of such differences and unwittingly select the wrong drivers. Jammed print queues lead to delays in printing throughput.

35 Another problem with conventional print managers is that they provide no guidance to assist in planning the positioning of printers within a building. Under-utilisation of printers may stem from in convenient positioning.

Furthermore, it would be desirable to remotely monitor print queues over a WAN and to have alerts from printers routed to a central monitoring point thereby enabling operators to identify printing problems centrally. Local site support can then be contacted with a view to resolving such problems.

If a large number of print jobs are allowed to build up in a print server, because a number of printers are off-line perhaps, then spooler directories in the print server may overflow thereby crashing the server.

In accordance with the present invention, there is now provided printer control apparatus for a server computer system of a data processing system comprising a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the apparatus comprising a receiver for receiving print jobs from the user terminal via the network, storage means for storing the print jobs in a print queue, a transmitter for sequentially sending the print jobs from the print queue to the printer via the network, and purging means for automatically deleting from the print queue print jobs stored in the print queue for a period in excess of a predetermined threshold.

Preferably, the storage means further comprises a print log for recording each print job received from the user terminal.

Examples of the present invention may further comprise means for generating an alert message on detection of a mismatch between a print job stored in the print queue and a print driver associated with the print job.

Preferred embodiments of the present invention may further comprise means for sending the alert message to the user terminal via the network. Alternatively, in some embodiments of the present invention, there may be provided means for sending the alert message to a network management terminal via the network. Also, in some embodiment of the present invention, there may be provided means for correcting the mismatch by automatically replacing the printer driver associated with the print job stored in the print queue.

It will be appreciated that the present invention extends to a server computer system comprising a central processing unit, a memory, printer control apparatus as hereinbefore described, and a bus



architecture interconnecting the central processing unit, the memory, and the printer control apparatus. It will also be appreciated that the present invention further extends to a data processing system comprising a user terminal, a printer, a server computer system as hereinbefore described, and a network interconnecting the user terminal, the printer, and the server computer system.

Viewing the present invention from another aspect, there is now provided a method for controlling a printer in a data processing system comprising a server computer system, a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the method comprising: receiving at the server print jobs from the user terminal via the network; storing the print jobs in a print queue in the server, sequentially sending the print jobs from the print queue in the server to the printer via the network; and, automatically deleting from the print queue in the server print jobs stored in the print queue for a period in excess of a predetermined threshold.

Viewing the present invention from yet another aspect, there is now provided a computer program product for a server computer system of a data processing system comprising a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the product comprising first code means for receiving print jobs from the user terminal via the network, second code means for storing the print jobs in a print queue, third code means for sequentially sending the print jobs from the print queue to the printer via the network, and fourth code means for automatically deleting from the print queue print jobs stored in the print queue for a period in excess of a predetermined threshold.

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a distributed data processing environment;

Figure 1 is a block diagram of a computer system of the data processing environment;

Figure 3, is a block diagram of a print server of the data processing environment;

Figure 4 is a block diagram of the print server in combination with a user terminal and a printer of the data processing environment; and,

Figure 5 is a flow diagram of a print manager portion of the print server.

Referring first to Figure 1 a distributed data processing system comprises a network 10. A plurality of user terminals 50-70 are connected to the network 10. A plurality of printers 20-40 are also connected to the network 10. Also connected to the network 10 are an application server computer system 80 and a print server computer system 90.

Referring to Figure 2, the printer server 90, application server 80, and the user terminals 50-70, each comprise a computer system having a random access memory (RAM) 110, a read only memory (ROM) 120, a central processing unit (CPU) 100, a mass storage device 130 comprising one or more large capacity magnetic disks or similar data recording media, a network adaptor 180, a keyboard adaptor 170, a pointing device adaptor 160, and a display adaptor 150 all interconnected via a bus architecture 140. A keyboard 200 is coupled to the bus architecture 140 via the keyboard adaptor 170. Similarly, a pointing device 210, such as a mouse, touch screen, tablet, tracker ball or the like, is coupled to the bus architecture 140 via the pointing device adaptor 160. Equally, a display output device 220, such as a cathode ray tube (CRT) display, liquid crystal display (LCD) panel, or the like, is coupled to the bus architecture 140 via the display adaptor 150. The bus architecture 140 is additionally coupled to the network 10 via the network adapter 180.

Basic input output system (BIOS) software is stored in the ROM 120 for enabling data communications between the CPU 100, mass storage 130, RAM 110, ROM 120, and the adaptors 150-180 via the bus architecture 140. Stored on the mass storage device 130 is operating system software and application software. The operating system software cooperates with the BIOS software in permitting control of the computer system by the application software.

In the mass storage of the application server 80 there are stored a plurality of application software packages such as word-processing packages and drawings packages for use by the user terminals 50-70.

5 Referring to Figure 3, the printer server 90 comprises a printer manager 260 for controlling supply of printing jobs from the user terminals 50-70 connected to the network 10 to the printers 20-40 also connected to the network 10. It will be appreciated that the print manager 260 may be stored in the mass storage 130 of the print server 90 for execution by the CPU 100 of the print server 90. The mass storage 130  
10 of the print server 90 also contains a plurality of print queues 230-250 each corresponding to a different one of the printers 20-40. A print log 270 is also maintained in the mass storage 130 of the print server 90.

15 With reference now to Figure 4, suppose for example that a user of user terminal 50 wishes to print, on printer 20, a document prepared on a word processor application loaded into user terminal 50 from the application server 80. The user specifies the desired printer, printer 20, when issuing the print request. The document is sent from the user terminal 50 to the print server via the network 20 as a print job 280  
20 with an accompanying print driver 290 corresponding to specified printer 20. At the print server 90, the print job 280 and driver 290 are temporarily stored in the queue associated with printer 20 until jobs already in the queue 230 are completed by printer 20. When the print job 280 reaches the head of the queue 230, it is sent to printer 20 for  
25 printing.

As mentioned earlier, the print manager 260 may be stored in the mass storage 130 of the print server 90 for execution by the CPU 100 of the print server 90. In a particularly preferred embodiment of the  
30 present invention, the print manager 260 is written in the C computer programming language. In operation, the printer manager 260 examines all incoming print and checks that they are using the correct driver for the intended printer 20-40. In the aforementioned particularly preferred  
35 embodiment, the print manager 260 comprises a REXX cmd file which is called by the C program when a problem is detected prompting an alert. The cmd file can be configured to send alerts to a central point using a proprietary protocol. The cmd file also deletes any print jobs remaining in a queue 230-250 beyond a predetermined time. Furthermore, the cmd file  
40 can be configured to send alerts to users and/or administrators in the event that an incorrect printer driver is specified.

In operation, the print manager 260 constantly monitors print queues 20-40 on the print server 90. Thus, the print manager can examine all jobs coming into each queue 230-250. In the aforementioned particularly preferred embodiment, the print manager 260 is configured by altering an INI file which determines the actions for the print manager 260 to take. Specifically, the INI file can be configured to: match a driver to a queue holding a corresponding job in the event that a mismatch is detected; send alerts relating to failing queues, jobs, and devices; delete any jobs in a queue that are over a specified age; and, log all print jobs in the print log 270.

What follows is a pseudo-code listing of the INI file showing the aforementioned functions provided by the print manager 260.

```

;PRINTMAN.INI
;This is the ini file that is called at initialisation of PRTMAN.EXE.
;This is the driver matching section
;MATCH: parameters:- yes or no
;This can be either yes or no. If yes it will match the JOB DRIVER to the
DEVICEDRIVERS.
;If the JOB DRIVER is unsupported the job will be held.
[DRIVERS]
match = yes

;This is the logging section
;log: parameters:- yes or no if yes all jobs are logged otherwise not
;use VLEWI.LOG.EXE to view the file, You need to stop printman by pressing
Q
;to do this
[LOGGING]
log = yes

;This is the NOTIFY section
;alertusers: parameters yes or no if yes users are sent network messages
otherwise not ;alertadmin: parameters
;yes or no if yes administrators are alerted otherwise not
;adminid: parameters id of an administrator use this to indicate an admin
id
;you wish to alert
[NOTIFY]
alertuser - yes

```

```

alertadmin = yes
adminid = alexl :

```

```

;This is the job count section. Enter the no of jobs to allow in a queue
;before sending an alert ;with the jobcount parameter. The timing of the
alert

```

```

;is altered with the jfreq parameter below.
;20 is the default forjobcount.

```

```

[NOJOBS]

```

```

jobcount = 20

```

```

;This is the alerting section.

```

```

;Parameters: yes or no

```

```

;Choosing yes enables the program to call PRTCMD.cmd a rexx cmd file that
can ;be used for example to send alerts remotely.

```

```

;queuealerts:-yes or no to turn on/off queuealerts

```

```

;qfreq:- no of queuealerts to ignore before alert is sent 4000 is the
default

```

```

;devicealerts:-yes or no to turn on/off devicealerts

```

```

;dfreq:-no of devicealerts to ignore before alert is sent 4000 is the
default

```

```

;jobalerts:-yes or no to turn on/off jobalerts

```

```

;jfreq:-no ofjobalerts to ignore before alert is sent 4000 is the default
[ALERTING]

```

```

enable - yes

```

```

queuealerts= yes

```

```

qfreq = 8000

```

```

devicealerts= yes

```

```

dfreq = 8000

```

```

jobalerts= yes

```

```

jfreq = 8000

```

```

;This is the scavenge section.

```

```

;If run is yes it will delete all jobs found on a print server that are
older ;than the specified kill time in hours. ktime will default to 12
hours if not ;specified & run = yes

```

```

[SCAVENGE]

```

```

run = yes

```

```

ktime =1?

```

Referring now to Figure 5, the print manager 260 is initially configured, at 300, via the aforementioned INI file. The print manager 260 operates by enumerating, at 310, all the print queues 230-250 on the print server 310. Then, the print manager 260 checks, at 320, the printers 20-40 attached to each queue 230-250. The print manager 260 follows the printer check 320 with a check, at 330, of all jobs waiting in the queues 230-250. In the event of a problem in a queue 230-250, a printer 20-40, or a job, a corresponding message is sent to a circular queue 340 in the print manager 260. The circular queue 340 is monitored by a separate thread 350 of the print manager 260. On detection of a message, the thread 350 calls PRTMAN.CMD, a rexx command file which, in turn, is sent the message as a parameter. PRTMAN.CMD can be configured via rexx to deal with the message. For example, PRTMAN.CMD may be configured to incorporate the message in a network message to a central administrator or a user via any proprietary protocol.

Referring back to Figure 3, in addition to fault reporting, the print manager 260 logs all jobs received in a log file 270. In particularly preferred embodiments of the present invention, the log file 270 is cyclical, holding of the order of 100000 jobs before recycling, or however many jobs are deemed sufficient to accommodate a year of average printing.

Returning to Figure 5, if the print manager detects, at 330, that any waiting job is over a predetermined age (eg: 12 hours, set in the INI file), the print manager 260 automatically deletes the job from the holding queue 230-250. The print manager 260 thus keeps the queue 230-250 clear of over-age jobs. In especially preferred embodiments of the present invention, the age threshold for purging jobs from the print queues 230-250 is adjustable by network administrators according to prevailing demands on the printers 20-40. Also, in especially preferred embodiments, the frequency with which queues are checked by the print manager 260 (eg: every 10 ms) is adjustable by network administration staff, again, according to prevailing demands on the printers 20-40.

It will be appreciated that the printer manager 260 hereinbefore described prevents print queues from jamming due to users specifying incorrect printer drivers. It will also be appreciated that the print manager 260 also prevents print queues 20-40 from overfilling with jobs by automatically deleting over-age jobs. Furthermore, it will be appreciated that, by logging incoming print jobs, the print manager 260

assists with determining the usage and location planning of printer installations. Still furthermore, the alerts generated by the print manager 260 enable printer problems to be quickly notified to a local or central administration points.

5

10

Preferred embodiments of the present invention have been hereinbefore described with reference to a data processing system in which the print server 260 is implemented in a separate computer system to the application server 80. However, it will be appreciated that the present invention is equally applicable to data processing systems in which the print server 90 and the application server 80 are integrated in the single computer.

15

20

25

In summary what has been hereinbefore described by way of example of the present invention is a printer controller 260 for a server computer system 90 of a data processing system comprising a user terminal 50-70, a printer 20-40, and a network 10 interconnecting the server 90, the user terminal 50-70 and the printer 20-40, which printer controller 260 receives print jobs from the user terminal 50-70 via the network 10, stores the print jobs in a print queue 230-250, sequentially sends the print jobs from the print queue 230-250 to the printer 20-40 via the network 10, and automatically deletes from the print queue 250-270 print jobs stored in the print queue 250-270 for a period in excess of a predetermined threshold.

CLAIMS

1. Printer control apparatus for a server computer system of a data processing system comprising a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the apparatus comprising a receiver for receiving print jobs from the user terminal via the network, storage means for storing the print jobs in a print queue, a transmitter for sequentially sending the print jobs from the print queue to the printer via the network, and purging means for automatically deleting from the print queue print jobs stored in the print queue for a period in excess of a predetermined threshold.
2. Apparatus as claimed in claim 1, wherein the storage means further comprises a print log for recording each print job received from the user terminal.
3. Apparatus as claimed in claim 1 or claim 2, comprising means for generating an alert message on detection of a mismatch between a print job stored in the print queue and a print driver associated with the print job.
4. Apparatus as claimed in claim 3, comprising means for sending the alert message to the user terminal via the network..
5. Apparatus as claimed in claim 3, comprising means for sending the alert message to a network management terminal via the network.
6. Apparatus as claimed in claim 3 comprising means for correcting the mismatch by automatically replacing the printer driver associated with the print job stored in the print queue.
7. A server computer system comprising a central processing unit, a memory, printer control apparatus as claimed in any preceding claim, and a bus architecture interconnecting the central processing unit, the memory, and the printer control apparatus.
8. A data processing system comprising a user terminal, a printer, a server computer system as claimed in claim 7, and a network interconnecting the user terminal, the printer, and the server computer system.



9. A method for controlling a printer in a data processing system comprising a server computer system, a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the method comprising: receiving at the server print jobs from the user terminal via the network; storing the print jobs in a print queue in the server, sequentially sending the print jobs from the print queue in the server to the printer via the network; and, automatically deleting from the print queue in the server print jobs stored in the print queue for a period in excess of a predetermined threshold.

10. A computer program product for a server computer system of a data processing system comprising a user terminal, a printer, and a network interconnecting the server, the user terminal and the printer, the product comprising first code means for receiving print jobs from the user terminal via the network, second code means for storing the print jobs in a print queue, third code means for sequentially sending the print jobs from the print queue to the printer via the network, and fourth code means for automatically deleting from the print queue print jobs stored in the print queue for a period in excess of a predetermined threshold.



Application No: GB 9724076.6  
Claims searched: 1-10

Examiner: Mike Davis  
Date of search: 12 February 1998

**Patents Act 1977**  
**Search Report under Section 17**

**Databases searched:**

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): G4H (HPC), G4A (AKB1)

Int Cl (Ed.6): G06F

Other:

**Documents considered to be relevant:**

Category	Identity of document and relevant passage	Relevant to claims
	None	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.